

Arduino - Entradas Analógicas

Por [Fábio Souza](#) - 18/11/2013



ÍNDICE DE CONTEÚDO

1. [Conversor A/D do Arduino](#)
2. [Funções da plataforma Arduino](#)
3. [Exemplo - Entradas Analógicas com Arduino](#)

No artigo anterior descrevemos os pinos de entrada e saída digitais abordando as suas características e as funções para trabalhar com escrita e leitura de sinais digitais na plataforma Arduino. Nesse artigo vamos aprender sobre as entradas analógicas e compreender seu funcionamento para no futuro aplicar essa técnica para seus projetos.

As entradas digitais só podem assumir dois estados, HIGH e LOW, ou seja, 0 V ou 5 V. Dessa forma só é possível ler apenas dois estados. Por exemplo, verificar se uma porta está aberta ou fechada, identificar se um botão está pressionado ou solto, etc. Com as entradas digitais você aplica em lógica discreta para controle de seus projetos, porém em muitas situações a variação das grandezas envolvidas acontece de forma analógica. Ou seja, variam continuamente em relação ao tempo e podem assumir infinitos valores dentro de uma faixa. Como exemplo a temperatura, pressão e umidade são grandezas que variam dessa forma.

O microcontrolador da Arduino trabalha internamente com dados digitais, portanto é necessário traduzir um sinal analógico para um valor digital. A técnica utilizada para leitura de um sinal analógico pelo Arduino é a conversão analógica digital. Essa técnica consiste em converter o sinal analógico para um valor digital, dessa forma se pode quantificar o sinal presente no pino. Esse processo é feito pelo conversor Analógico digital, ADC ou conversor A/D.

Um conversor A/D quantifica o valor analógico conforme a quantidade de bits da sua resolução. A resolução de um conversor A/D é dada pela seguinte equação:

$$\text{resolução} = V_{ref} / 2^n$$

onde:

Vref: tensão de referência do conversor A/D;

n: número de bits do conversor.

1. Conversor A/D do Arduino

O conversor A/D do microcontrolador ATmega328 possui 10 bits de resolução, a sua tensão de entrada pode variar de 0 V até o valor de VCC e possui referência interna selecionável de 1,1 V.

Dessa forma quando está trabalhando com a referência em VCC o menor valor que pode ser lido será:

$$\text{Resolução} = 5 \text{ V} / 1024 = 4,88 \text{ mV}$$

esse é o valor de degrau para uma conversão em 10 bits com referência em 5V.

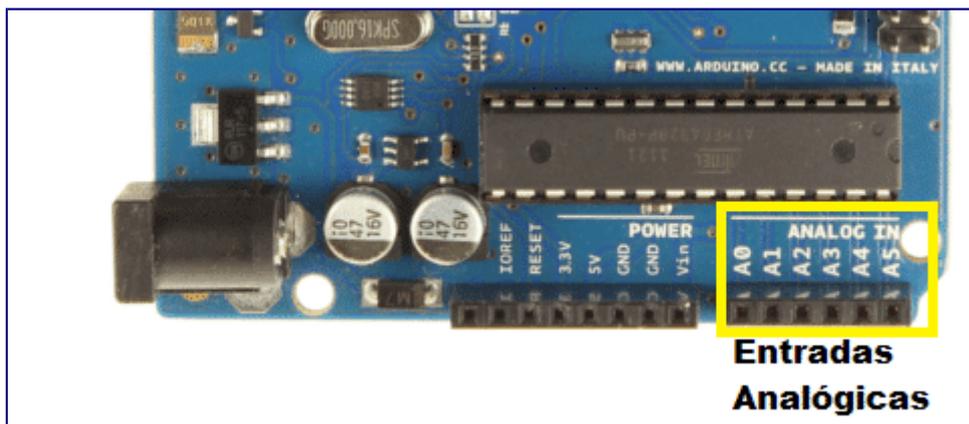
Caso trabalhe com a referência interna de 1,1V a resolução será:

$$\text{resolução} = \frac{1,1}{1024} = 1,07 \text{ mV}$$

Nota-se que o passo é bem menor para esse valor de referência.

Se a referência externa for selecionada, a resolução dependerá do valor de tensão aplicada ao pino AREF.

A placa Arduino UNO possui 6 canais de conversor analógico digital. Essas entradas são nomeadas de A0 a A5 e são exibidas na figura a seguir:



2. Funções da plataforma Arduino

A plataforma Arduino possui funções para trabalhar com entradas analógicas, as quais abstraem a configuração dos registradores de configuração do conversor A/D, assim como selecionam o canal conforme o pino passado como parâmetro. São apenas duas funções e são exibidas a seguir:

- **analogReference(tipo)**

Descrição

Configura a referência de tensão para a conversão analógica/digital, usando esse valor como o máximo para a entrada analógica.

Os tipos possíveis de configurações são:

- **DEFAULT:** a tensão padrão para conversão é a tensão de alimentação da placa. 5 V para placas alimentadas com 5 V e 3,3 V para placas alimentadas com 3,3 V;
- **INTERNAL:** referência interna de 1,1V no Atmega168 e Atmega328, e 2,56 V no ATmega8;
- **INTERNAL1V1:** referência de 1,1V, apenas no Arduino Mega;
- **INTERNAL2V56:** referência interna de 5,6 V, apenas no Arduino Mega;
- **EXTERNAL:** referência de tensão aplicada no pino AREF (valor entre 0 e 5V).

Sintaxe:

`analogReference(tipo);`

Parâmetros:

tipo: DEFAULT, INTENAL, INTERNAL1V1, INTERNAL2V56, EXTERNAL.

Retorno:

Essa função não tem retorno algum.

- **int analogRead(pino)**

Descrição

Lê o valor presente em um pino configurado como entrada analógica. Internamente o Arduino possui um conversor A/D de 10 bits. Dessa forma o valor retornado por esta função estará na faixa de 0 a 1023 conforme o valor presente no pino.

O tempo para leitura pela função `analogRead()` é por volta de 100 micro segundos, dessa forma a máxima frequência de leitura que se pode ter é de 10000 vezes por segundo.

Sintaxe:

```
analogRead(pino);
```

Parâmetros:

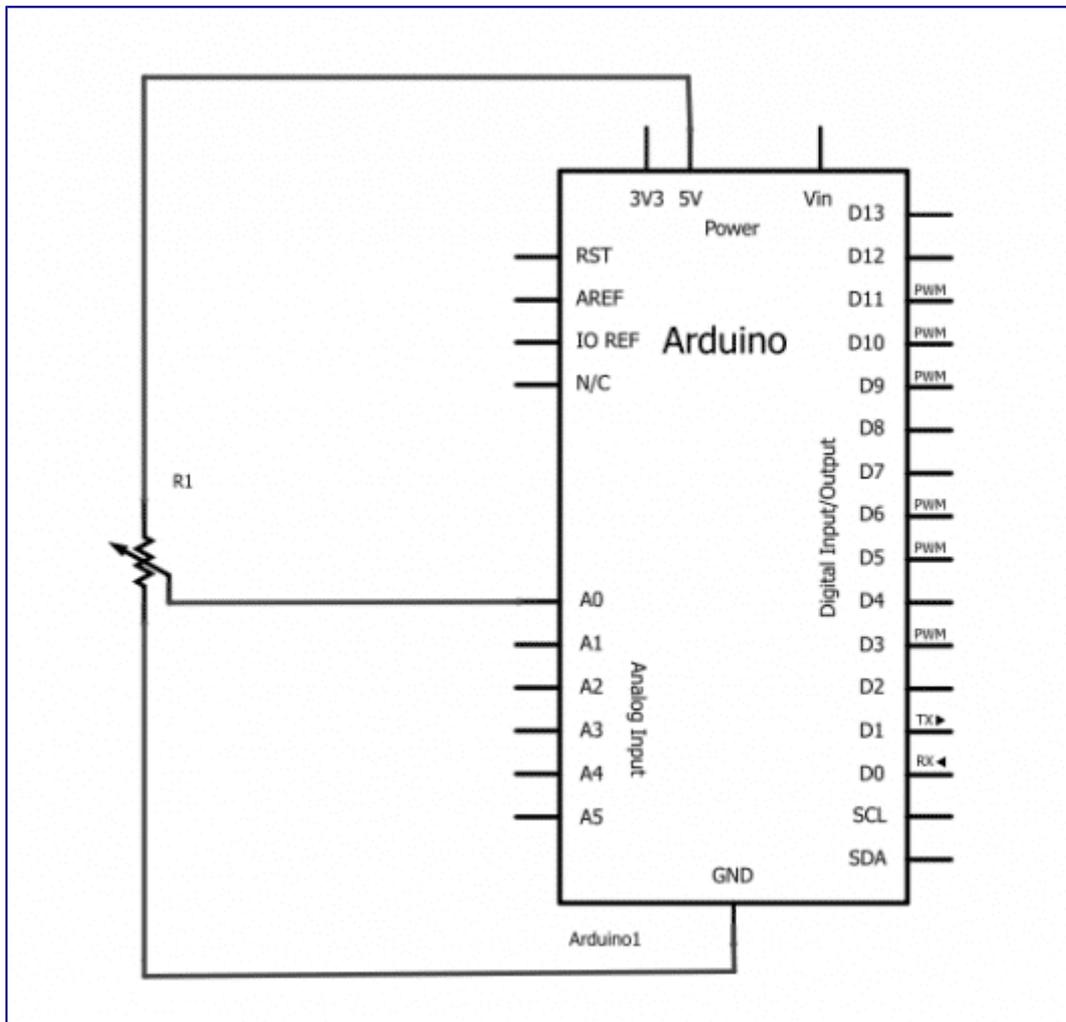
pino: valor do pino configurado como entrada analógica (0 a 5 na maioria da placas, 0 a 7 na MINI e NANO, 0 a 15 na MEGA).

Retorno

int (0 a 1023)

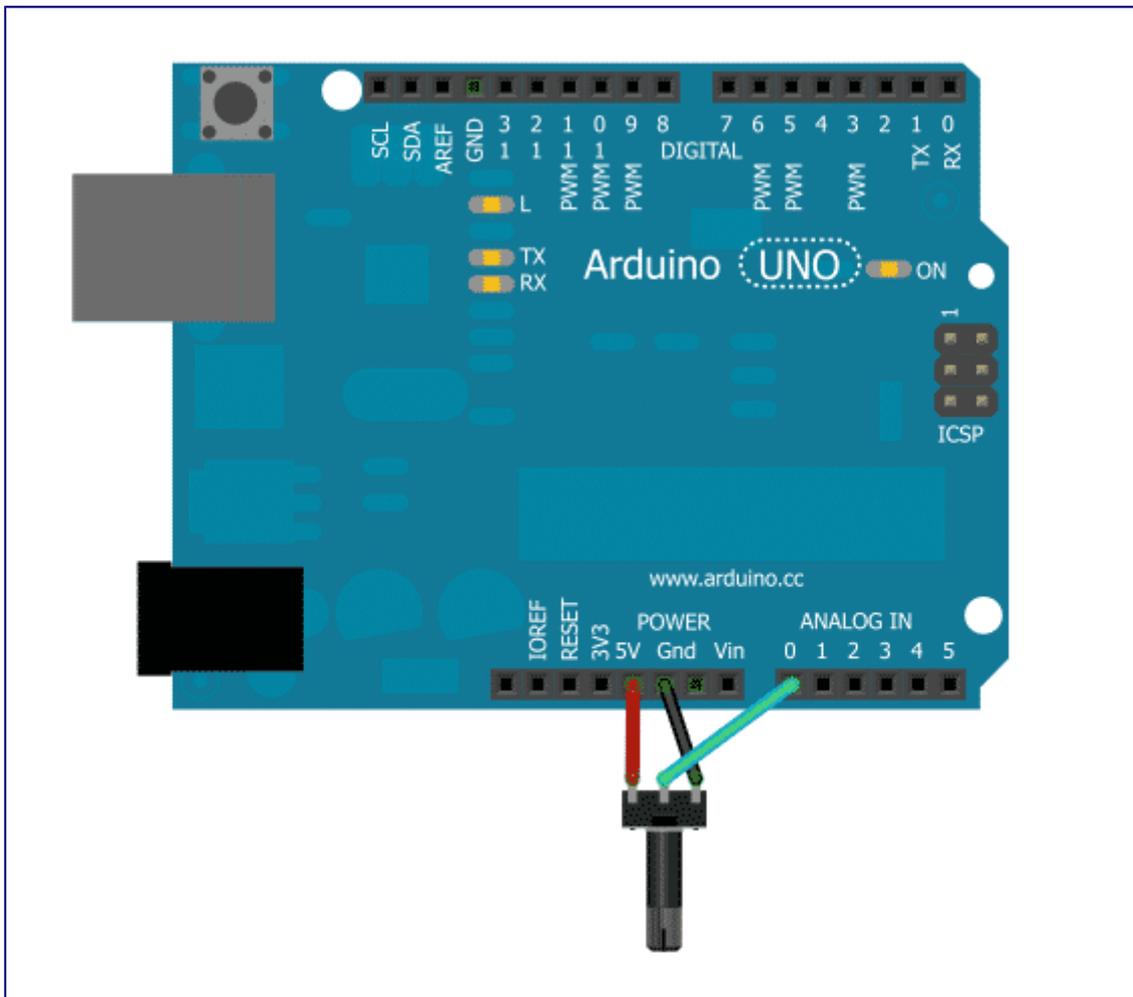
3. Exemplo - Entradas Analógicas com Arduino

Para iniciar o estudo da entrada analógica, a maneira mais simples e rápida é ligando um potenciômetro a uma entrada analógica, conforme esquema apresentado em seguida:



Se girarmos o cursor do potenciômetro, alteramos a resistência em cada lado do contato elétrico que está conectada ao terminal central do botão. Isso provoca a mudança na proximidade do terminal central aos 5 volts ou GND, o que implica numa mudança no valor analógico de entrada. Quando o cursor for levado até o final da escala, teremos, por exemplo, 0 V a ser fornecido ao pino de entrada do Arduino e, assim, ao lê-lo obtém-se 0. Quando giramos o cursor até o outro extremo da escala, haverá 5 V a ser fornecido ao pino do Arduino e, ao lê-lo, teremos 1023. Em qualquer posição intermediária do cursor, teremos um valor entre 0 e 1023, que será proporcional à tensão elétrica sendo aplicada ao pino do Arduino.

A ligação no Arduino UNO pode ser feita conforme a figura abaixo:



O exemplo a seguir lê o valor no potenciômetro. O tempo que o LED permanece ligado ou desligado depende do valor obtido pelo `analogRead()`.

```

1  /*
2  * Entrada analógica
3  * Liga e desliga um LED conectado ao pino digital 13. O tempo
4  * que o LED permanece ligado ou desligado depende do valor
5  * obtido pelo analogRead().
6  */
7
8
9  int potPin = 0; // selecione o pino de entrada ao potenciômetro
10 int ledPin = 13; // selecione o pino ao LED
11 int val = 0; // variável a guardar o valor proveniente do sensor
12
13
14 void setup() {
15
16   pinMode(ledPin, OUTPUT); // declarar o pino ledPin como saída
17
18 }
19
20
21
22 void loop() {
23
24   val = analogRead(potPin); // ler o valor do potenciômetro
25
26   digitalWrite(ledPin, HIGH); // ligar o led
27   delay(val); // espera tempo ajustado no potenciometro
28   digitalWrite(ledPin, LOW); // desligar o led
29   delay(val); // espera tempo ajustado no potenciometro
30
31 }

```

O exemplo apresentado tem estrutura bem simples e serve para fixar os conceitos sobre o conversor A/D do Arduino. Aproveitaremos essa estrutura para artigos futuros onde faremos a leituras de sensores.

Fonte: <https://www.embarcados.com.br/arduino-entradas-analogicas/>